IN THE SPECIFICATION

Please replace the paragraph on Page 1, beginning on line 8, with the following amended paragraph.

An IC card is a device which keeps tamper-prohibited personal information ~~which is not allowed to tamper~~ or performs encryption of data or decryption of a ciphertext with the use of secret keys. An IC card itself does not have its own power supply, and when it is inserted into a reader/writer for an IC card, power is supplied to the IC card and it is made operable. After it is made operable, the IC card receives commands transmitted from the reader/writer, and following the commands the IC card processes, for example, transfer of data. A general explanation of an IC card is given in Junichi Mizusawa, "IC card", ~~Ohm-sha, denshi-tsuushin-jouhou-gakkai-hen~~ Ohm Corporation, compiled by the Society of Electronic Communication and Information, etc.

Please replace the paragraph on Page 16, beginning on line 2, with the following amended paragraph.

As a method of calculation, an addition chain method or the like is often adopted (refer to "Angouriron Nyuumon" ("An Introduction to the Theory of Cryptography")); however, with such an algorithm processing speed is slow, and the time required for a transaction utilizing an IC card may exceed the user's allowable time. Therefore, it is the CRT to produce M from the result of a modular exponentiation for 2 prime factors, P and Q, of the public modulus N, instead of simply performing the modular exponentiation for X and N.

Please replace the paragraph on Page 22, beginning on line 21, with the following amended paragraph.

Note that the above-mentioned addition chain method and sliding window method can be executed with the use of a technique called the Montgomery method. The Montgomery method is one to be used for the high speed execution of the modular multiplication arithmetic operation, AB MOD N. In particular, it is suited to implementation by hardware. The algorithm of the method will be briefly explained. Details are described in Montgomery's ~~literally~~ work, "Modular Multiplication Without

Trial Division", Mathematics of Computation 44, 170, pp. 519 to 521 (1985).

Please replace the paragraph on Page 42, lines 11-24, with the following amended paragraph.

For the convenience of explanation, it is assumed that ALU 1605 is an arithmetic operator of 8 bits width, and the width of each of bus lines, 1606, 1607 and 1608 is 16 bits. Explanation will be made for two registers. The register 1610 is used on the source side, and the register 1613 is used on the destination side. The ~~register 1610 has~~ registers 1610, 1613 each have the capacity of 16 bits and ~~it is~~ are composed of HIGH side 8 bits ~~1609~~ 1609, 1612 and LOW side 8 bits ~~1611,~~ 1611, 1614, respectively. There is provide a CCR (Condition Code Register) 1615 for storing flags: flags showing the results of operation; for example, a 0 flag which shows 1 when the value of the result is 0, and which shows 0 if the result is not 0, or a carry-flag which shows carry up. The CCR is connected to the ALU 1605 and to the various kinds of buses (1606, 1607, 1608).

Please replace the paragraph on Page 45, beginning on line 14 through Page 47, line 4, with the following amended paragraph.

In the following, we put: A = (A[N-1]A[N-2] . . . A[1]A[-[0], B = (B[N-1]B[N-2] . . . B[1]B[0] (each of A[J] and B[J] is a 16-bit block. Fig. 22 shows the procedures of calculating the sum of A and B. At first, A and B are received (step 1801). In this place, "receive" means not only to receive the signals through the I/O port; but also to have A and B determined as a result of other calculation. Next, the counter J is initialized to 0 (step 1802). The conditional branch process (step 1803) judges if the counter is N or not. When J = N, it means that all bit blocks have been processed, so that the processing is put to an end. If J is not N, the process is advanced to step 1804. At step 1804, 1 bit random number V is generated. After that, at the conditional branch process of step 1805, whether V is 0 or 1 is judged. In the case of 0, A[J] is transferred to the source register RS (step 1806). After that, B[J] is transferred to the destination register RD (step 1807). After that, ADD.W RS, RD is executed (step 1810). At step 1810, the sum of the values of RS and RD is calculated, and the value is transferred to RD. When a carry up occurs, it is held as a carry, and in the process of the next ADD.W, the process of adding 1 is executed. Since this portion is processed by the instruction of ADD.W, it is not written in the flow chart. A programmer of assembler language does not usually handle the processing of carry. Next, the contents of RD are stored onto the designated position C[J] on the RAM (step 1811). Next, J is incremented by 1 (step 1812), and the procedure is returned to the conditional branch process of the step 1803. Inversely, when V is 1, B[J] is transferred to the destination register RD (step 1808). After that, A[J] is transferred to the source register RS (step 1809). After that, ADD.W RS, RD is executed (step 1810). At step 1810, the sum of the values of RS and RD is calculated, and the value is transferred to RD. If a carry up occurs, it is held as a carry, and in the process of the next ADD.W process, the process of adding 1 is executed. Next, the contents of RD [[is]] are stored to the designated position C[J] on the RAM (step 1811). Next, J is incremented by 1 (step 1812), and the procedure moves back to the conditional branch process of step 1803.

Please replace the paragraph on Page 49, beginning on line 11 through Page 51, line 2, with the following amended paragraph.

Fig. 24 and Fig. 25 show the procedures of calculating the product of A and B. At first, A and B are received (step 2001). In this place the word "receive" means not only to receive the signals through the I/O port; but also to have A and B determined as a result of other calculation. Next, the counters I and J are initialized to 0 (step 2002). The conditional branch process (step 2003) judges whether the counter J is N or not. If J = N, it means all bit blocks have been processed, so that the procedure is brought to an end. If J is not N, the procedure is advanced to step 2004. At step 2004, 1 bit random number V is generated. After that the conditional branch process (step 2005) judges if V is 0 or 1. If it is 0, A[I] is transferred to the source register RSL (step 2006), and after that B[J] is transferred to the destination register RDL (step 2007). After that, MULTI RSL, RD is executed (step 2010), and the value of the destination register RD in which the value of the partial product is stored is transferred to the temporary storage area TMP1 on the RAM (step 2011). Next, the counter is incremented by 1 (step 2012), and the process is advanced to the conditional branch process (step 2013). The conditional branch process (step 2013) judges if the counter I is N. If I = N, the counter j is incremented by 1 and I is initialized to 0 (step 2025), and the process procedure is returned to the conditional branch (step 2003) to perform operations as described above. At the conditional branch process (step 2013), if I is not N, 1 bit random number V [[of]] is generated (step ~~2015~~ 2014), then V is checked (step 2015), and if V = 0, after A[1] is transferred to RSL (step 2016), B[J] is transferred to RDL (step 2017). Inversely, if V = 1, after B[J] is transferred to RDL (step 2018), A[I] is transferred to RSL (step 2019). After processing of either step, step 2017 or step 2019, MULTI RSL, RD is executed (step 2020), and RD is transferred to TMP2 (STEP 2021). Then the value of TMP2 is shifted to the left by 8 bits (step 2022), the sum of TMP1 and

TMP2 is calculated (on this operation refer to the explanation in Fig. 21), and the result is transferred to TMP1 (step 2023). Then I is incremented by 1 (step 2024), and the procedure is returned to the conditional branch (step 2013).

Please replace the paragraph on Page 51, beginning on lines 3, with the following amended paragraph.

In the above process, a random number is generated and according to the value of it, the process is switched; either first A[J] is transferred to RS and then B[J] is transferred to RD, or inversely first B[J] is transferred to RD and then A[J] is transferred to RS. With the above-mentioned switching, the waveform of the consumption current of an IC chip is varied[[,]]; in particular, if one tries a process of averaging the waveforms to eliminate noises for observing the difference of data (for the typical oscilloscope, noises are eliminated by this method), the waveform is observed only as the mean value of A and B; therefore [[and]] it becomes difficult to estimate the contents of each of them. Although ~~not showing~~ an example is not shown, it is easy to limit the present embodiment to the sum of a single 8-bit block. In the present embodiment, the conditional branch process is switched with a random number [[v]] V, but is easy to change it to a pseudo-random number using a linear congruence method, a chaotic sequence or a predetermined bit pattern, and the above alternatives have nothing to do with the essentials of the present invention.

Please replace the paragraph on Page 57, beginning on line 23, with the following amended paragraph.

In the present embodiment, to utilize the embodiments shown in FIG. 22 and FIG. 26 for the change of order of the process in step 2506 and the process in step 2507, will help ~~improve~~ produce more ~~effect~~ improvement than to use the present invention independently.